REMARKS

Reconsideration of this application is respectfully requested.

RESTRICTION REQUIREMENT

Claims 1-16 are presented for examination. In response to the Examiner's restriction requirement applicants elect Group I for examination and cancel claims 17-21 without prejudice.

CLAIM REJECTIONS

Rejections under 35 U.S.C. §102(b)

The Examiner has rejected claims 1-16 under 35 U.S.C. §102(b) as being anticipated by Mattson et al. (U.S. Patent 5,717,893) (hereafter, "Mattson"). Applicants submit that claims 1-16 are not anticipated by Mattson. In regard to the rejection of claims 1 and 7, the Examiner has stated in part that:

> Mattson teaches the invention as claimed including an apparatus and method for dynamically partitioning a cache array based upon requests for memory from an integrated device having a plurality of processors... (5/3/02, Office Action, p. 4)

Applicants respectfully submit that claim 1 is not anticipated by Mattson. Claim 1 recites the feature of "partitioning a cache array dynamically..." (Emphasis added) Mattson does not disclose this feature as can be seen by the following analysis of Mattson. Mattson discloses a method for managing a cache hierarchy. (Mattson, title) The way that Mattson manages a cache is that when an application requests a block of data, B, the cache manager must find B, tag B with a DataType, Tb, make room for it in partition P0 of the cache, and make it the Most Recently Used block in P0. (Mattson, col. 8, ll. 17-21). In Mattson, the cache manager is instructed to change the partition sizes at predetermined times in an attempt to maximize the number of hits to a cache. (Mattson, col. 9, lines 60-62) (Emphasis added).

Mattson further describes the dynamic categorization of blocks of data into disjoint DataTypes. (Mattson, col. 10, ll. 5-20). Thus, Mattson discloses a method of organizing data hierarchically in a cache array at *predetermined* times. However, because Mattson does not disclose "partitioning a cache array dynamically based upon requests for memory..." as taught by claim 1, applicants respectfully submit that claim 1 is not anticipated under 35 U.S.C. §102(b) by Mattson.

Furthermore, because Mattson does not disclose this feature as taught by applicants and given that claims 2-6 depend directly or indirectly from claim 1, applicants respectfully submit that claims 1-6 are not anticipated under 35 U.S.C. §102(b) by Mattson.

The Examiner also rejected independent claim 7 under 35 U.S.C. §102(b) for the reason set forth in the rejection of claim 1. Claim 7 discloses substantially similar limitations as claim 1, and recites "a cache memory array dynamically partitioned when multiple memory requests are received...." (Emphasis added) Because, Mattson does not disclose this feature as taught by applicants for the reasons discussed above with regard to claim 1, applicants respectfully submit that claim 7 is not anticipated under 35 U.S.C. §102(b) by Mattson.

Furthermore, because Mattson does not disclose the feature of "a cache memory array dynamically partitioned when multiple memory requests are received...," as taught by applicants in independent claim 7 from which claims 8-10 depend, applicants respectfully submit that claims 7-10 are not anticipated under 35 U.S.C. §102(b) by Mattson.

Rejection Under 35 U.S.C. §103

The Examiner also rejected independent claim 11 under 35 U.S.C. §103(a) as being unpatentable over Mattson and alleged knowledge in the art. Claims 11-16 are patentable under 35 U.S.C. S. 103 in view of the references cited by the Examiner. Neither the cited reference, nor the alleged knowledge in the art teach (nor does the Office Action cite any portion which even

suggests) the presently claimed feature of partitioning a cache array dynamically based upon requests for memory.

The Examiner states in part that "one of ordinary skill in the art would have recognized that a computer readable medium is well-known in the art." (5/3/02 Office Action, p. 6) Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. *In re Fine*, 837 F.2d 1071 (Fed. Cir. 1988).

In regard to the rejection of claim 11, even if Mattson and the alleged knowledge in the art were combined, such a combination would lack one or more features of claim 11 from which claims 12-16 directly or indirectly depend. Claim 11 recites the feature of partitioning a cache array dynamically based upon requests for memory from an integrated device having a plurality of processors. (Emphasis added) As discussed above with regard to the rejection of claim 1, Mattson nor the alleged knowledge in the art disclose this feature.

Mattson discloses a method for managing a cache hierarchy. (Mattson, title) The way that Mattson manages a cache is that when an application requests a block of data, B, the cache manager must find B, tag B with a DataType, Tb, make room for it in partition P0 of the cache, and make it the Most Recently Used block in P0. (Mattson, col. 8, ll. 17-21). In Mattson, the cache manager is instructed to change the partition sizes at predetermined times in an attempt to maximize the number of hits to a cache. (Mattson, col. 9, lines 60-62) (Emphasis added).

Mattson further describes the dynamic categorization of blocks of data into disjoint DataTypes. (Mattson, col. 10, ll. 5-20). Thus, Mattson discloses a method of organizing data hierarchically in a cache array at *predetermined* times. However, because Mattson does not disclose "partitioning a cache array dynamically based upon requests for memory..." as taught by claim 11, it is respectfully submitted that claim 11 and dependent claims 12-15 are patentable under 35 U.S.C. §103(a) over Mattson and the alleged knowledge in the art.

It is respectfully submitted that in view of the arguments set forth herein, the applicable rejections have been overcome.

If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: July 26, 2002

Sanjeet K. Dutta
Reg. No. 46,145

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 947-8200

VERSION OF CLAIMS WITH MARKINGS TO SHOW CHANGES MADE:

CLAIMS

We claim:

1. A method, comprising:

partitioning a cache array dynamically based upon requests for memory from an integrated device having a plurality of processors.

2. The method as claimed in claim 1, further comprising subdividing one or more ways within the cache array.

3. The method as claimed in claim 1, further comprising subdividing one or more sets within the cache array.

4. The method as claimed in claim 1, further comprising using a single least recently used array to replace ways.

5. The method as claimed in claim 1, further comprising applying a multiple pseudo least recently used update based on an entry hit.

6. The method as claimed in claim 1, further comprising partitioning dynamically the cache array into a direct-mapped cache.

7. A device comprising:

a cache memory array dynamically partitioned when multiple memory requests are received from an integrated device having a plurality of processors.

8. The device as claimed in claim 7 further comprising:

an integrated device having a plurality of processors connected to the cache memory array.

9. The device as claimed in claim 7 further comprising a main memory device connected to the cache memory array.

10. The device as claimed in claim 8 wherein the integrated device includes a graphics processor and a central processing unit.

11. A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform the method of:

partitioning a cache array dynamically based upon requests for memory from an integrated device having a plurality of processors.

12. The computer-readable medium of claim 11 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform the method of subdividing one or more ways within the cache array.

13. The computer-readable medium of claim 11 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform the method of subdividing one or more sets within the cache array.

14. The computer-readable medium of claim 11 having stored thereon-additional instructions, said additional instructions when executed by a computer, cause said computer to further perform the method of using a single least recently used array to replace ways.

15. The computer-readable medium of claim 11 having stored thereon-additional instructions, said additional instructions when executed by a computer, cause said computer to further perform the method of applying a multiple pseudo least recently used update based on an entry hit.

16. The computer-readable medium of claim 11 having stored thereon-additional instructions, said additional instructions when executed by a computer, cause said computer to further perform the method of partitioning dynamically the cache array into a direct-mapped cache.

17. (Cancelled) [A method, comprising:

converting an N-way set associative cache dynamically into a direct mapped cache; including

removing M least significant bits from a tag address, and

adding the M least significant bits to M most significant bits of a set address of the direct-mapped cache.]

18. (Cancelled) [The method of claim 17, wherein N equals 2 to the power M.]

19. (Cancelled) [A method, comprising:

converting an N-way set associative cache dynamically into a Z x N-way set associative cache; including

providing Y+1 virtual copies of a pseudo-LRU array for the N-way set associative cache, wherein the pseudo-LRU array is not replicated, and

selecting a virtual copy with Y most significant bits of a set address for the N-way set associative cache.]

20. (Cancelled) [The method of claim 19, wherein Z is 2 to the power Y, where Y is greater than or equal to 1.]

21. (Cancelled) [The method of claim 19, wherein the Y most significant bits of the set address for the N-way set associative cache become the Y least significant bits of the tag address for the Z x N-way set associative cache.]